

(19)日本国特許庁 (JP)

## (12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-346154

(43)公開日 平成11年(1999)12月14日

(51)Int.Cl.<sup>a</sup>

H 03 M 7/14

識別記号

F I

H 03 M 7/14

B

審査請求 未請求 請求項の数31 OL (全24頁)

(21)出願番号 特願平10-150280

(22)出願日 平成10年(1998)5月29日

(71)出願人 000002185  
ソニー株式会社  
東京都品川区北品川6丁目7番35号(71)出願人 598070935  
フィリップス・エレクトロニクス・エヌ・  
ブイ  
Koninklijke Philips  
Electronics NV.  
オランダ アイントホーフェン市 ビイ・  
エイ5621 フローネフォウセ通り 1

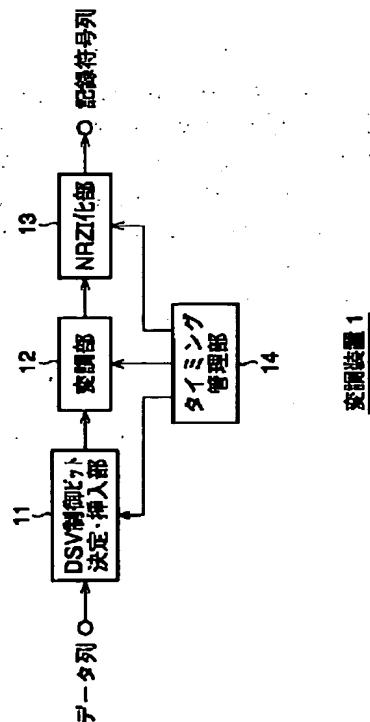
(74)代理人 弁理士 稲本 義雄

最終頁に続く

(54)【発明の名称】変調装置および方法、復調装置および方法、並びに提供媒体

(57)【要約】

【課題】高線密度での記録再生ができるようにする。  
 【解決手段】DSV制御ビット決定・挿入部11は、入力されたデータ列にDSV制御のためのDSV制御ビットを挿入し、変調部12に出力する。変調部12は、変換テーブルに従って、基本データ長が2ビットのデータを、基本符号長が3ビットの可変長符号に変換して、NRZI化部13に出力する。変調部12が有する変換テーブルは、最小ランの連続を所定の回数以下に制限する置き換えコード、ラン長制限を守るための置き換えコードを有し、さらに、データ列の要素内の「1」の個数を2で割ったときの余りと、符号語列の要素内の「1」の個数を2で割ったときの余りが、どちらも1あるいは0で一致するような変換規則を有する。



## 【特許請求の範囲】

【請求項1】 基本データ長がmビットのデータを、基本符号長がnビットの可変長符号(d, k; m, n; r)に変換する変調装置において、  
入力されたデータを、変換テーブルに従って、符号に変換する変換手段を備え、  
前記変換テーブルの変換コードは、  
 $d = 1, k = 7, m = 2, n = 3$ の基礎コードと、  
データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、  
最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、  
ラン長制限を守るための第2の置き換えコードとを有することを特徴とする変調装置。

【請求項2】 拘束長i=1における前記基礎コードを構成するデータ列と符号語列の対の数は、 $2^m = 2^2 = 4$ よりも少ないことを特徴とする請求項1に記載の変調装置。

【請求項3】 前記変換テーブルの基礎コードは、可変長構造を有することを特徴とする請求項1に記載の変調装置。

【請求項4】 前記変換テーブルの基礎コードは、不確定符号を含み、前記不確定符号は、直前または直後の符号語が「1」のとき「0」となり、「0」のとき「1」となる記号を\*とするとき、「000」または「101」となる符号「\*0\*」を含むことを特徴とする請求項1に記載の変調装置。

【請求項5】 前記変換テーブルの変換コードは、直後の符号語列、または、直後に続くデータ列を参照して決定するコードを含むことを特徴とする請求項1に記載の変調装置。

【請求項6】 前記参照する直後の符号語列は、特定の1種類の符号語列とすることを特徴とする請求項5に記載の変調装置。

【請求項7】 前記直後の符号語列、または、直後に続くデータ列を参照して決定する変換コードは、前記第1または第2の置き換えコードであることを特徴とする請求項5に記載の変調装置。

【請求項8】 拘束長iが1である場合における前記基礎コードを構成するデータ列と符号語列の対の数は、 $2^m = 2^2 = 4$ に等しいことを特徴とする請求項1に記載の変調装置。

【請求項9】 拘束長iが2以上の場合の前記変換コードは、すべて前記第1または第2の置き換えコードであることを特徴とする請求項1に記載の変調装置。

【請求項10】 前記置き換えコードは、拘束長iが2の場合の変換コードは、最小ランd=1を守るコードであることを特徴とする請求項1に記載の変調装置。

【請求項11】 前記変換テーブルの変換コードは、直前の符号語列を参照して決定するコードを含むことを特徴とする請求項10に記載の変調装置。

【請求項12】 前記変換テーブルの変換コードとして存在しないユニークなパターンを含む同期信号を、前記符号語列の任意の位置に挿入する挿入手段をさらに備えることを特徴とする請求項1に記載の変調装置。

【請求項13】 前記同期信号に用いられるユニークなパターンは、最大ランkを破るパターンであることを特徴とする請求項12に記載の変調装置。

【請求項14】 前記同期信号に用いられるユニークなパターンは、最小ランdを守るパターンであることを特徴とする請求項12に記載の変調装置。

【請求項15】 前記同期信号は、先頭の1符号語が、直前までのデータを変換した符号語との接続ビットであり、2番目の符号語が、最小ランdを守るためのビットであり、3番目の符号語から、前記同期信号としてユニークなパターンを構成することを特徴とする請求項12に記載の変調装置。

【請求項16】 前記同期信号の大きさは、少なくとも12符号語であることを特徴とする請求項12に記載の変調装置。

【請求項17】 前記同期信号は、その大きさが、21符号語以上のとき、最大ランk=8のパターンを少なくとも2個含むことを特徴とする請求項12に記載の変調装置。

【請求項18】 前記変換テーブルの変換コードは、符号を任意の位置において終端させるための終端コードをさらに有することを特徴とする請求項12に記載の変調装置。

【請求項19】 前記終端コードは、前記データ列と符号語列の対の数が、 $2^m = 2^2 = 4$ よりも少ない前記拘束長iの前記基礎コードに対応して規定され、かつ、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則を有することを特徴とする請求項18に記載の変調装置。

【請求項20】 前記終端コードを識別するために、前記同期信号パターンにおいて接続ビットとされた、先頭の1符号語ビットを、前記終端コードを用いたときには「1」とし、そうでないときは「0」とすることを特徴とする請求項18に記載の変調装置。

【請求項21】 前記同期信号は、その先頭の3ビット、及び、最後の3ビットが、データ列及び接続用の混合した接続ビットとされ、その間が、前記ユニークなパターンとされることを特徴とする請求項12に記載の変調装置。

【請求項22】 前記同期信号の先頭の3ビットは、変換前のデータ語において、mビット単位で見て、先頭の

1ビット目をデータ語に対応する値とし、次の2ビット目を前記同期信号を規定するために「1」とし、前記同期信号の最後の3ビットは、変換前のデータ語において、mビット単位で見て、先頭の1ビット目は同期信号を規定するために「0」とし、次の2ビット目をデータ語に対応する値とすることを特徴とする請求項12に記載の変調装置。

【請求項23】 入力されたデータのDSVを制御して、前記変換手段に供給するDSV制御手段をさらに備えることを特徴とする請求項1に記載の変調装置。

【請求項24】 前記変換手段は、前記最小ランdの連続を制限する前記第1の置き換えコードを検出する第1の検出手段と、前記ラン長制限を守る前記第2の置き換えコードを検出する第2の検出手段とを備えることを特徴とする請求項1に記載の変調装置。

【請求項25】 基本データ長がmビットのデータを、基本符号長がnビットの可変長符号(d, k ; m, n ; r)に変換する変調装置の変調方法において、入力されたデータを、変換テーブルに従って、符号に変換する変換ステップを含み、

前記変換テーブルの変換コードは、  
 $d = 1, k = 7, m = 2, n = 3$  の基礎コードと、  
 データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、  
 最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、  
 ラン長制限を守るための第2の置き換えコードとを有することを特徴とする変調方法。

【請求項26】 基本データ長がmビットのデータを、基本符号長がnビットの可変長符号(d, k ; m, n ; r)に変換する変調装置に、  
 入力されたデータを、変換テーブルに従って、符号に変換する変換ステップを含む処理を実行させるプログラムを提供する提供媒体であって、  
 前記変換テーブルの変換コードは、

$d = 1, k = 7, m = 2, n = 3$  の基礎コードと、  
 データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、  
 最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、  
 ラン長制限を守るための第2の置き換えコードとを有することを特徴とする提供媒体。

【請求項27】 基本符号長がnビットの可変長符号(d, k ; m, n ; r)を、基本データ長がmビットのデータに変換する復調装置において、

入力された符号を、変換テーブルに従って、データに変換する変換手段を備え、

前記変換テーブルの変換コードは、

$d = 1, k = 7, m = 2, n = 3$  の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、

最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、

ラン長制限を守るための第2の置き換えコードとを有することを特徴とする復調装置。

【請求項28】 所定の間隔で挿入された冗長ビットを除去する除去手段をさらに備えることを特徴とする請求項27に記載の復調装置。

【請求項29】 前記冗長ビットは、DSV制御ビット、または同期信号であることを特徴とする請求項28に記載の復調装置。

【請求項30】 基本符号長がnビットの可変長符号(d, k ; m, n ; r)を、基本データ長がmビットのデータに変換する復調装置の復調方法において、入力された符号を、変換テーブルに従って、データに変換する変換ステップを含み、

前記変換テーブルの変換コードは、  
 $d = 1, k = 7, m = 2, n = 3$  の基礎コードと、  
 データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、

最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、

ラン長制限を守るための第2の置き換えコードとを有することを特徴とする復調方法。

【請求項31】 基本符号長がnビットの可変長符号(d, k ; m, n ; r)を、基本データ長がmビットのデータに変換する復調装置に、

入力された符号を、変換テーブルに従って、データに変換する変換ステップを含む処理を実行させるプログラムを提供する提供媒体であって、

前記変換テーブルの変換コードは、  
 $d = 1, k = 7, m = 2, n = 3$  の基礎コードと、  
 データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、

最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、

ラン長制限を守るための第2の置き換えコードとを有することを特徴とする提供媒体。

【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、変調装置および方法、復調装置および方法、並びに提供媒体に関し、特に、データを、記録媒体に、高密度に記録または再生する場合に用いて好適な変調装置および方法、復調装置および方法、並びに提供媒体に関する。

## 【0002】

【従来の技術】データを所定の伝送路に伝送したり、または例えば磁気ディスク、光ディスク、光磁気ディスク等の記録媒体に記録する際、伝送路や記録媒体に適するように、データの変調が行われる。このような変調方法の1つとして、ブロック符号が知られている。このブロック符号は、データ列を $m \times i$ ビットからなる単位（以下データ語という）にブロック化し、このデータ語を適当な符号則に従って、 $n \times i$ ビットからなる符号語に変換するものである。そしてこの符号は、 $i = 1$ のときには固定長符号となり、また $i$ が複数個選べるとき、すなわち、1乃至 $i_{\max}$ （最大の*i*）の範囲の所定の*i*を選択して変換したときには可変長符号となる。このブロック符号化された符号は可変長符号（ $d, k; m, n; r$ ）と表される。

【0003】ここで*i*は拘束長と称され、 $i_{\max}$ は*r*（最大拘束長）となる。また*d*は、連続する”1”的間に入る、”0”的最小連続個数、例えば”0”的最小ランを示し、*k*は連続する”1”的間に入る、”0”的最大連続個数、例えば”0”的最大ランを示している。

【0004】ところで上述のようにして得られる可変長符号を、光ディスクや光磁気ディスク等に記録する場合、例えばコンパクトディスク（CD）やミニディスク（MD）では、可変長符号を、”1”を反転とし、”0”を無反転として、NRZI（Non Return to Zero Inverted）変調し、NRZI変調された可変長符号（以下、記録波形列とも称する）に基づき記録が行なわれている。また、記録密度のあまり大きくなかった初期のISO規格の光磁気ディスクでは、記録変調されたビット列が、NRZI変調されず、そのまま記録されていた。

【0005】記録波形列の最小反転間隔を $T_{min}$ とし、最大反転間隔を $T_{max}$ とするとき、線速方向に高密度に記録を行うためには、最小反転間隔 $T_{min}$ は長い方が、すなわち最小ラン*d*は大きい方が良く、またクロックの再生の面からは、最大反転間隔 $T_{max}$ は短い方が、すなわち最大ラン*k*は小さい方が望ましく、この条件を満足するために、種々の変調方法が提案されている。

【0006】具体的には、例えば光ディスク、磁気ディスク、又は光磁気ディスク等において、提案あるいは実際に使用されている変調方式として、可変長符号であるRLL（1-7）（(1, 7; m, n; r)とも表記される）やRLL（2-7）（(2, 7; m, n; r)とも表記される）、そしてISO規格MOに用いられている固定長RLL（1-7）（(1, 7; m, n; 1)とも表記

される）などがある。現在開発研究されている、記録密度の高い光ディスクや光磁気ディスク等のディスク装置では、最小ラン*d* = 1のRLL符号（Run Length Limited Code）がよく用いられている。

【0007】可変長RLL（1-7）符号の変換テーブルは、例えば以下のようなテーブルである。

&lt;表1&gt;

RLL (1, 7 ; 2, 3 ; 2)

データ	符号
$i = 1$	1 1      0 0 x
	1 0      0 1 0
	0 1      1 0 x
$i = 2$	0 0 1 1      0 0 0 0 0 x
	0 0 1 0      0 0 0 0 1 0
	0 0 0 1      1 0 0 0 0 x
	0 0 0 0      1 0 0 0 1 0

【0008】ここで変換テーブル内の記号xは、次に続くチャネルビットが”0”であるときには”1”とされ、また次に続くチャネルビットが”1”であるときに”0”とされる。最大拘束長*r*は2である。

【0009】可変長RLL（1-7）のパラメータは（1, 7 ; 2, 3 ; 2）であり、記録波形列のビット間隔をTとすると、 $(d+1) T$ で表される最小反転間隔 $T_{min}$ は $2 (= 1 + 1) T$ となる。データ列のビット間隔をTdataとすると、この $(m/n) \times 2$ で表される最小反転間隔 $T_{min}$ は $1.33 (= (2/3) \times 2) T_{data}$ となる。また $(k+1) T$ で表される最大反転間隔 $T_{max}$ は $8 (= 7 + 1) T (= (m/n) \times 8 T_{data} = (2/3) \times 8 T_{data} = 5.33 T_{data})$ である。さらに検出窓幅 $T_w$ は $(m/n) \times T_{data}$ で表され、その値は $0.67 (= 2/3) T_{data}$ となる。

【0010】ところで、表1のRLL（1-7）による変調を行ったチャネルビット列においては、発生頻度としては $T_{min}$ である $2 T$ が一番多く、以下 $3 T$ 、 $4 T$ と続く。 $2 T$ や $3 T$ のようなエッジ情報が早い周期で多く発生するのは、クロック再生には有利となる場合が多い。

【0011】ところが、さらに記録線密度を高くしていくと、今度は逆に、最小ランが問題となってくる。すなわち最小ランである $2 T$ が連続して発生し続けると、記録波形に歪みが生じやすくなってくる。なぜなら、 $2 T$ の波形出力は、他の波形出力よりも小さく、例えばデフォーカスやタンジェンシャル・チルト等による影響を受け易いからである。またさらに、高線密度記録の際にには、最小マーク（ $2 T$ ）の連続した記録はノイズ等の外乱の影響も受け易くなり、従ってデータ再生時に誤りが起こり易くなる。この場合におけるデータ再生誤りのパターンとしては、連続する最小マークの先頭と最後がシフトして誤るケースが多く、その結果、発生するピットエラー長が長くなってしまうことになる。

【0012】ところで、記録媒体へのデータの記録、あ

るいは、データの伝送の際には、記録媒体あるいは伝送路に適した符号化変調が行われるが、これら変調符号に直流成分が含まれていると、例えば、ディスク装置のサポーの制御におけるキャッシングエラーなどの、各種のエラー信号に変動が生じ易くなったり、あるいはジッターが発生し易くなったりする。従って、変調符号には、直流成分をなるべく含めないようにする方が良い。

【0013】そこで、DSV(Digital Sum Value)を制御することが提案されている。このDSVとは、チャネルビット列をNRZI化し(すなわちレベル符号化し)、そのビット列(データのシンボル)の”1”を”+1”、”0”を”-1”として、符号を加算していくときのその総和を意味する。DSVは符号列の直流成分の目安となる。DSVの絶対値を小さくすること、すなわち、DSV制御を行うことは、符号列の直流成分を抑制することになる。

【0014】上記表1に示した、可変長RLL(1-7)テーブルによる変調符号は、DSV制御が行われていない。このような場合のDSV制御は、変調後の符号化列(チャネルビット列)において、所定の間隔でDSV計算を行い、所定のDSV制御ビットを符号化列(チャネルビット列)内に挿入することで、実現する。

【0015】しかしながら、DSV制御ビットは、基本的には冗長ビットである。従って符号変換の効率から考えれば、DSV制御ビットはなるべく少ない方が良い。

【0016】またさらに、挿入されるDSV制御ビットによって、最小ランdおよび最大ランkは、変化しない方が良い。(d, k)が変化すると、記録再生特性に影響を及ぼしてしまうからである。

#### 【0017】

【発明が解決しようとする課題】以上のように、RLL符号を高線密度にディスクに記録再生する場合、最小ランdの連続したパターンがあると、長いエラーが発生し易いという課題があった。

【0018】また、RLL(1-7)符号のようなRLL符号において、DSV制御を行う場合には、符号語列(チャネルビット列)内に、任意の間隔で、DSV制御ビットを入れる必要があった。DSV制御ビットは冗長であるから、なるべく少い方が望ましいが、最小ランあるいは最大ランを守るために、DSV制御ビットが少なくとも2ビット以上必要であり、DSV制御ビットをより短くすることが望まれている。

【0019】本発明は、このような状況に鑑みてなされたものであり、最小ランd=1であるRLL符号(d, k; m, n)=(1, 7; 2, 3)において、最小ランの連続する回数を制限し、さらに最小ラン及び最大ランを守りながら、効率の良い制御ビットで、DSV制御を行うことができるようすることを目的とする。

【0020】また本発明は、なるべく単純な構造の変換テーブルを用いて、復調エラー伝搬が増大するのを抑制するようにすることを目的とする。

#### 【0021】

【課題を解決するための手段】請求項1に記載の変調装置は、入力されたデータを、変換テーブルに従って、符号に変換する変換手段を備え、変換テーブルの変換コードは、d=1, k=7, m=2, n=3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0022】請求項2-5に記載の変調方法は、入力されたデータを、変換テーブルに従って、符号に変換する変換ステップを含み、変換テーブルの変換コードは、d=1, k=7, m=2, n=3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0023】請求項2-6に記載の提供媒体は、基本データ長がmビットのデータを、基本符号長がnビットの可変長符号(d, k; m, n; r)に変換する変調装置に、入力されたデータを、変換テーブルに従って、符号に変換する変換ステップを含む処理を実行させるプログラムを提供する提供媒体であつて、変換テーブルの変換コードは、d=1, k=7, m=2, n=3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0024】請求項2-7に記載の復調装置は、入力された符号を、変換テーブルに従って、データに変換する変換手段を備え、変換テーブルの変換コードは、d=1, k=7, m=2, n=3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0025】請求項3-0に記載の復調方法は、入力された符号を、変換テーブルに従って、データに変換する変換ステップを含み、変換テーブルの変換コードは、d=1, k=7, m=2, n=3の基礎コードと、データ列

の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0026】請求項31に記載の提供媒体は、基本符号長がnビットの可変長符号(d, k; m, n; r)を、基本データ長がmビットのデータに変換する復調装置に、入力された符号を、変換テーブルに従って、データに変換する変換ステップを含む処理を実行させるプログラムを提供する提供媒体であって、変換テーブルの変換コードは、d=1, k=7, m=2, n=3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0027】請求項1に記載の変調装置、請求項25に記載の変調方法、および請求項26に記載の提供媒体、並びに、請求項27に記載の復調装置、請求項30に記載の復調方法、および請求項31に記載の提供媒体においては、データ列の要素内の「1」の個数と、符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則の変換コード、最小ランdの連続を所定の回数以下に制限する第1の置き換えコード、および、ラン長制限を守るための第2の置き換えコードに基づいて、変換処理が行われる。

#### 【0028】

【発明の実施の形態】以下に本発明の実施の形態を説明するが、特許請求の範囲に記載の発明の各手段と以下の実施の形態との対応関係を明らかにするために、各手段の後の括弧内に、対応する実施の形態(但し一例)を付加して本発明の特徴を記述すると、次のようになる。但し勿論この記載は、各手段を記載したものに限定することを意味するものではない。

【0029】請求項1に記載の変調装置は、入力されたデータを、変換テーブル(例えば、表2)に従って、符号に変換する変換手段(例えば、図1の変調部12)を備え、変換テーブルの変換コードは、d=1, k=7, m=2, n=3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、

<表2>

17PP.RML.32

データ	符号
11	*0*
10	001

どちらも1あるいは0で一致するような変換規則と、最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0030】請求項12に記載の変調装置は、変換テーブルの変換コードとして存在しないユニークなパターンを含む同期信号を、符号語列の任意の位置に挿入する挿入手段(例えば、図9の同期信号挿入部212)をさらに備えることを特徴とする。

【0031】請求項23に記載の変調装置は、入力されたデータのDSVを制御して、変換手段に供給するDSV制御手段(例えば、図1のDSV制御ビット決定・挿入部11)をさらに備えることを特徴とする。

【0032】請求項24に記載の変調装置は、変換手段は、最小ランdの連続を制限する第1の置き換えコードを検出する第1の検出手段(例えば、図3の最小ラン連続制限コード検出部33)と、ラン長制限を守る第2の置き換えコードを検出する第2の検出手段(例えば、図3の最大ラン補償コード検出部34)とを備えることを特徴とする。

【0033】請求項27に記載の復調装置は、入力された符号を、変換テーブル(例えば、表2)に従って、データに変換する変換手段(例えば、図5の復調部111)を備え、変換テーブルの変換コードは、d=1, k=7, m=2, n=3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ランdの連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0034】請求項28に記載の復調装置は、所定の間隔で挿入された冗長ビットを除去する除去手段(例えば、図5のDSV制御ビット除去部112)をさらに備えることを特徴とする。

【0035】次に、本発明の実施の形態について説明するが、以下においては、説明の便宜上、変換される前のデータの「0」と「1」の並び(変換前のデータ列)を、「000011」のように、( )で区切って表し、変換された後の符号の「0」と「1」の並び(符号語列)を、「000100100」のように、「 」で区切って表すこととする。以下に示す表2及び表3は、本発明のデータを符号に変換する変換テーブルの例を表している。

#### 【0036】

```

01          010
0011        010 100
0010        010 000
0001        000 100
000011      000 100 100
000010      000 100 000
000001      010 100 100
000000      010 100 000
"110111     001 000 000(next 010)
00001000    000 100 100 100
00000000    010 100 100 100
if xx1 then *0* = 000
xx0 then *0* = 101
-----
```

## Sync &amp; Termination

#01 000 000 001 (12 channel bits)

or

#01 001 000 000 001 000 000 001 (24 channel bits)

# = 0 not terminate case

# = 1 terminate case

## Termination table

00 000

0000 010 100

"110111 001 000 000(next010):

When next channel bits are '010',  
convert '11 01 11' to '001 000 000' after  
using main table and termination table.

【0037】表2の変換テーブルは、変換コードとして、それがないと変換処理ができない基礎コード（データ列（11）から（000000）までのコード）、それがなくても変換処理は可能であるが、それがあると、より効果的な変換処理が可能となる置き換えコード（データ列（110111），（00001000），（00000000）のコード）、および、符号を任意の位置で終端させるための終端コード（データ列（00），（0000）のコード）を含んでいる。また、この変換テーブルには、同期信号も規定されている。

【0038】また、表2は、最小ランd = 1、最大ランk = 7で、基礎コードの要素に不確定符号（\*を含む符号）を含んでいる。不確定符号は、直前および直後の符号語列の如何によらず、最小ランdと最大ランkを守るように、"0"か"1"に決定される。すなわち表2において、変換する2ビットのデータ列が（11）であったとき、その直前の符号語列によって、"000"または"101"が選択され、そのいずれかに変換される。すなわち、直前の符号語列の1チャネルビットが"1"である場合、最小ランdを守るために、2ビットのデータ（11）は、符号語"000"に変換され、直前の符号語列の1チャネルビットが"0"である場合、最大ラン

ランkが守られるように、符号語"101"に変換される。

【0039】表2の変換テーブルの基礎コードは可変長構造を有している。すなわち、拘束長i = 1における基礎コードは、必要数の4つ ( $2^m = 2^2 = 4$ ) よりも少ない3つ (\*0\*, 001, 010の3つ) で構成されている。その結果、データ列を変換する際に、拘束長i = 1だけでは変換出来ないデータ列が存在することになる。結局、表2において、全てのデータ列を変換するには（変換テーブルとして成り立つためには）、拘束長i = 3までの基礎コードを参照する必要がある。

【0040】また、表2の変換テーブルは、最小ランdの連続を制限する置き換えコードを持っているため、データ列が（110111）である場合、さらに後ろに続く符号語列が参照され、それが"010"であるとき、このデータ列は、符号語"001 000 000"に置き換えられる。また、このデータ列は、後ろに続く符号語列が"010"以外である場合、2ビット単位（(11), (01), (11)）で符号語に変換されるので、符号語"\*0\* 010 \*0\*"に変換される。これによって、データを変換した符号語列は、最小ランの連続が制限され、最大でも6回までの最小ラン繰り返し

となる。

【0041】さらに、表2の変換コードは、データ列の要素内の”1”的個数を2で割った時の余りと、変換される符号語列の要素内の”1”的個数を2で割った時の余りが、どちらも1あるいは0で同一（対応するいずれの要素も、”1”的個数が奇数または偶数）となるような変換規則を持っている。例えば、変換コードのうちのデータ列の要素(000001)は、”010100100”的符号語列の要素に対応しているが、それぞれの要素の”1”的個数は、データ列では1個、対応する符号語列では3個であり、どちらも2で割ったときの余りが1（奇数）で一致している。同様にして、変換コードのうちのデータ列の要素(000000)は、”010100000”的符号語列の要素に対応しているが、それぞれ”1”的個数は、データ列では0個、対応する符号語列では2個であり、どちらも2で割ったときの余りが0（偶数）で一致している。

【0042】そして、表2では、最大拘束長 $r=4$ である。拘束長 $i=4$ のコードは、最大ラン $k=7$ を実現するための置き換えコード（最大ラン補償コード）で構成されている。すなわち、データ(000010000)は、符号語”000100100100”に変換され、データ(00000000)は、符号語”010100100100”に変換されるようになされている。なお、この場合にも、最小ラン $d=1$ は守られている。

【0043】この拘束長 $i=4$ の置き換えコードを設けない時、表2は最大拘束長 $r=3$ となり、最大ラン $k$ が8の符号を作ることができる。しかし、この拘束長 $i=4$ のコードを設けることで、最大ラン $k$ を7にすることができる。一般的に、最大ラン $k$ が大きいほど、クロックの再生には不利となり、システムの安定性が悪くなる。従って、最大ラン $k$ を8から7にすることで、この特性がそれだけ改善されることになる。

【0044】以上より表2のテーブルにおいて、基礎コードのみによってテーブルを構成する場合は、最大拘束長 $r=3$ となり、最小ラン $d=1$ で最大ラン $k=8$ 、かつデータ列の要素内の”1”的個数を2で割った時の余りと、変換される符号語列の要素内の”1”的個数を2で割った時の余りが、どちらも1あるいは0で同一となるような、符号を発生することができる。

【0045】また基礎コードに加えて、最小ラン $d$ の連続を制限する置き換えコードを持って構成する場合は、最大拘束長 $r=3$ となり、最小ラン $d=1$ で最大ラン $k=8$ 、かつ最小ラン $d$ の連続が有限回までに制限され、さらにデータ列の要素内の”1”的個数を2で割った時の余りと、変換される符号語列の要素内の”1”的個数を2で割った時の余りが、どちらも1あるいは0で同一となるような、符号を発生することができる。

【0046】そして、表2のように、基礎コードに加えて、最小ラン $d$ の連続を制限する置き換えコードを持

ち、さらに最大ラン $k$ を7に補償する置き換えコードを持って構成する場合は、最大拘束長 $r=4$ となり、最小ラン $d=1$ で最大ラン $k=7$ 、かつ最小ラン $d$ の連続が有限回までに制限され、さらにデータ列の要素内の”1”的個数を2で割った時の余りと、変換される符号語列の要素内の”1”的個数を2で割った時の余りが、どちらも1あるいは0で同一となるような、符号を発生することができる。

【0047】また、一般的に、最大拘束長 $r$ が大きいほど、ビットシフト時の復調エラー（エッジビットの位置が1ビット分だけ、正規の位置よりも前方または後方にシフトすることによるエラー）の伝搬特性が悪くなる。

【0048】但し、表1と表2を比較すると、表1の最大拘束長 $r$ は2であるのに対して、表2の最大拘束長 $r$ は4である。従って、表1より表2の方がこの特性が悪いはずである。しかしながら、表7を参照して、シミュレーション結果として後述するように、表2の場合のこの特性は、表1のそれに較べてそれ程、劣るものではなかった（表7に示すように、平均バイトエラーレートが、表1の場合、1.014バイトであるのに対して、表2の場合、1.167バイトとなっており、それ程大きくなっていない）。これは、変換コードの対の数が2個と少ないためと考えられる。

【0049】ところで上記表2の変換テーブルによって発生された符号語列（チャネルビット列）中の、任意の位置に同期信号を挿入する場合、この変換テーブルは可変長構造を有しているために、任意の位置で符号を終端させるために終端用テーブルが規定され、必要に応じて用いられるようになされている。

【0050】例えば、任意の位置で同期信号を挿入する際、まず直前直後の符号語列との接続において、最小ラン $d$ 及び最大ラン $k$ が守られるように接続ビットが設定され、接続ビットの間に同期信号用のユニークなパターンが設定される。同期信号パターンとして、最大ラン $k=7$ を破るパターンを与えたとき、最も短い長さで実現できる同期信号パターンは、次に示すように、12符号語（12チャネルビット）である。

”#0100000001”

この同期信号パターンの先頭の”#”は接続用ビットで、”0”か”1”的どちらかに設定される。”#”の次の第2チャネルビット目は、最小ランを守るために”0”に設定される。第3チャネルビット目から、同期信号パターンとして、 $k=8$ となる9Tのユニークなパターン（表2には規定されていないコードのパターン）が設定される。すなわち”1”と”1”的間に、”0”が8個連續して並ぶ。なおこの同期信号パターンの最後のチャネルビットは”1”とされているが、表2の変換テーブルを用いた時は、このようにしても、最小ラン $d$ を守ることができる。

【0051】次に終端用テーブルと、この同期信号バタ

ーンの接続用ビット”#”について説明する。終端用テーブルは、表2に示すように、

```
00 000
0000 010 100
```

となる。終端用テーブルが必要になるのは、データ列と符号語列の対の数が4つ ( $2^m = 2^2 = 4$ ) よりも少ない拘束長*i*の基礎コードに対してである。

【0052】すなわち、表2では、拘束長*i* = 1における基礎コードのデータ列と符号語列の対の数は3つであるから終端用テーブルが必要となる。また拘束長*i* = 2における基礎コードのデータ列と符号語列の対の数も3つであるから終端用テーブルが必要となる。拘束長*i* = 3における変換コードのデータ列と符号語列の対の数は5つあり、そのうち1つが置き換えコードで、4つが基礎コードであり、必要数(4個)を持っているので、この場合、終端用テーブルは不要となる。拘束長*i* = 4におけるコードは、いずれも置き換えコードであるため、符号の終端を考慮しなくてよい。従って、終端用テーブルは、*i* = 1の(00)のときと、*i* = 2のときの(0000)のときに必要になる。この終端用テーブルにより、データ(00)は、符号”000”に変換され、データ(0000)は、符号”010100”に変換され

”#01 000 000 001

のように、*k* = 8 (9T) のパターンを2回繰り返すことができ、このようにすると、同期信号をより確実に検

”#01 001 000 000

のように、”3T - 9T - 9T”を同期信号パターンとし、2回連続する*k* = 8 (9T) のパターンの前後に、大きなラン(T)が来る確率を減らし、さらに検出能力を高くすることができる。どの程度の検出能力の同期信号を採用するかは、システムの要求によって、選択する

<表3>

17PP.RML.52

*i*=1 Main table:

データ	符号
00	101
01	100
10	001
11	000

*i*=2 substitution table A. (limits d to 1)

0000	100 010
0001	101 010
1000	000 010
1001	001 010

*i*=3 substitution table B. (limits k to 8)

111111	000 010 010
111110	001 010 010
011110	101 010 010
011111	100 010 010

*i*=4 substitution table C. (limits RMTR to 6)

る。これにより、同期信号を挿入するに際し、その直前のデータを符号に変換することができなくなる(同期信号の直前までの符号を終端させることができなくなる)ことが防止される。

【0053】同期信号パターンの接続用ビット”#”は、終端用のテーブルを用いる場合と、用いない場合を区別するためのものである。すなわち、同期信号として与えられた、先頭の第1チャネルビット目の”#”は、終端コードを用いたときは「1」とされ、そうでないときは「0」とされる。こうすることによって、テーブルの違い(終端コードを用いたか否か)を、間違なく識別することができる。

【0054】以上においては、同期信号パターンを最短である12符号語(12チャネルビット)として説明したが、同期信号パターンとしては、最大ラン*k* = 7を破る、*k* = 8 (9T) が作成できれば良いので、12符号語以上であれば同期信号パターンがこの他にも作れることがある。例えば15符号語ならば

”#01 000 000 001 010”

”#01 000 000 001 001”

のように、2通りの同期信号を作ることができる。また、21符号語ならば、

000 000 001”

出することが可能となる。そして、24符号語の場合、

001 000 000 001”

が出来る。

【0055】表3は、本発明の他の変換テーブルの例を示している。

【0056】

```

00010001 100 010 010 010
chan --0 10010001 100 000 010 010
chan --1 10010001 000 010 010 010
r=4 substitution table D. (limits k to 7)
chan 010 11100000 000 001 010 010
chan 010 11100010 100 001 010 010
chan 010 11100001 001 010 010 010
chan 010 11100011 101 010 010 010

```

## Sync

```

data: x1.....0x
ch. : xx0 100 000 000 10x (12channel bits)
data: x1.....0x
ch. : xx0 100 000 000 100 000 000 10x (24channel bits)

```

## Termination :

```

add data bits '01' or '11' at begin,
and '00' or '01' at the ent

```

【0057】表3は、最小ラン $d = 1$ 、最大ラン $k = 7$ で、拘束長 $i = 1$ において基礎コードを4つ( $2^m = 2^2 = 4$ )持つ構造をしている。すなわち、表3では、拘束長 $i = 1$ の場合がメインテーブルとされ、拘束長 $i$ が、2以上の場合は、 $d$ 、 $k$ 等を制限するための置き換えコードのテーブルとなっている。すなわち、 $i = 2$ のテーブルAでは、最小ラン $d$ を1に制限する置き換えコードが規定されており、 $i = 3$ のテーブルBでは、最大ラン $k$ を8までに制限する置き換えコードが規定されており、さらに、 $i = 4$ のテーブルC及びテーブルDでは、最小ラン $d = 1$ の連続を制限する置き換えコード(テーブルC)と、最大ラン $k$ を7までに制限する置き換えコード(テーブルD)とが規定されている。表3では、最大拘束長 $r = 4$ である。

【0058】このように、表3は、変換テーブル内に、最小ランの連続を制限する置き換えコードを持っているので、例えば、データ列(00010001)は、符号語列"100010010010"に置き換えられる。また、データ列(10010001)は、その直前の符号語列を参照して、それが"0"か"1"かによって置き換えコードが選択され、"0"ならば、"100000010010"に、"1"ならば、"000010010010"に、それぞれ変換される。これによって、データ変換後の符号語列は、最小ランの連続が制限され、最大でも6回までの最小ラン繰り返しの符号語列となる。

【0059】また、表3は、データ列の要素内の「1」の個数を2で割ったときの余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも、1あるいは0で、同一となるような変換規則を持っている。例えば、データ列の要素(1000)は、"000010"の符号語列に対応しているが、それぞれ

の「1」の個数は、データ列では1個(奇数)、対応する符号語列では1個(奇数)であり、どちらも2で割ったときの余りが1(奇数)で一致している。同様に、データ列の要素(1111111)は"000010010"の符号語列に対応しているが、それぞれの"1"の個数は、データ列では6個(偶数)、対応する符号語列では2個(偶数)であり、どちらも2で割ったときの余りが0(偶数)で一致している。

【0060】さらに、表3は、最大拘束長である $r = 4$ のコードにおいて、最大ラン $k = 7$ を実現するための置き換えコードを持つ。このとき、その直前の符号語列が参照され、それが、"010"であるとき置き換えが実行される。例えば、データ(11100000)は、直前の符号語が"010"であれば、符号語"000001010010"に変換され、データ(11100010)は、直前の符号語が"010"であれば、符号語"1000001010010"に変換される。

【0061】以上より表3のテーブルにおいて、RLL符号を実現するためには、基礎コードのみによってテーブルを構成することはできない。

【0062】最小ランおよび最大ランを補償するために、基礎コードおよび、テーブルA(拘束長 $i = 2$ )、テーブルB( $i = 3$ )を用いることで、RLL符号を作成することができる。この場合は、最大拘束長 $r = 3$ となり、最小ラン $d = 1$ で最大ラン $k = 8$ 、かつデータ列の要素内の"1"の個数を2で割った時の余りと、変換される符号語列の要素内の"1"の個数を2で割った時の余りが、どちらも1あるいは0で同一となるような、符号を発生することができる。

【0063】また基礎コード、テーブルA及びテーブルBに加えて、最小ラン $d$ の連続を制限する置き換えコード(テーブルC)を持って構成する場合は、最大拘束長

$r = 4$  となり、最小ラン  $d = 1$  で最大ラン  $k = 8$ 、かつ最小ラン  $d$  の連続が有限回までに制限され、さらにデータ列の要素内の”1”的個数を2で割った時の余りと、変換される符号語列の要素内の”1”的個数を2で割った時の余りが、どちらも1あるいは0で同一となるような、符号を発生することができる。なおこの場合では、表3内のテーブルCのような、直前の符号語列を参照することは、必ずしも必要ではない。

【0064】そして、表3のように、基礎コード、テーブルA、Bに加えて、最小ラン  $d$  の連続を制限する置き換えコード（テーブルC）を持ち、さらに最大ラン  $k$  を7に補償する置き換えコード（テーブルD）を持って構成する場合は、最大拘束長  $r = 4$  となり、最小ラン  $d = 1$  で最大ラン  $k = 7$ 、かつ最小ラン  $d$  の連続が有限回までに制限され、さらにデータ列の要素内の”1”的個数を2で割った時の余りと、変換される符号語列の要素内の”1”的個数を2で割った時の余りが、どちらも1あるいは0で同一となるような、符号を発生することができる。

【0065】上記表3の変換テーブルを利用して発生させた符号語列（チャネルビット列）中の任意の位置に同期信号を挿入する場合、この変換テーブルは拘束長  $i = 1$  において終端が可能であるから、表2で述べたような終端用テーブルは不要である。

【0066】そして、同期信号パターンをなるべく効率良く挿入するために、以下のように同期信号パターンを決定する。すなわち同期信号としてのユニークなパターン（符号語列）の前後の3符号語を、表3に従って変換された符号語で構成されるようにする。直前直後の3ビットは、以下に示すように、データビットと接続ビットが混合した形式となっている。

【0067】すなわち、直前の3ビット（符号語）は、変換前のデータ語において、 $m$ ビット（2ビット）単位で見て、先頭の1ビット目を情報データ語とし、次の2ビット目を同期信号を規定するために「1」とし、これ

” $xx0\ 100\ 000\ 000$ ”

のように、 $k = 8$  (9T) のパターンを2回繰り返すパターンを同期信号パターンとすることができます。このような同期信号パターンにすれば、同期信号検出能力を強めることができる。どの程度の検出能力のパターンを同期信号パターンとするかは、システムの要求によって、選択することができる。

【0071】表2及び表3のような変換テーブルを用いた時、従来の場合と同様に、データ列を変調した後、変調後のチャネルビット列に、所定の間隔で、DSV制御ビットを付加することで、DSV制御することができる。しかしながら、表2および表3においては、データ列と、変換される符号語列の関係を生かして、さらに効率良くDSV制御を行うことができる。

【0072】即ち、変換テーブルが、データ列の要素内

を変換テーブル表3において符号語（チャネルビット）に変換したものとする。このとき変換前の $m$ ビット（2ビット）のデータ語（ $x1$ ）は、 $n$ ビット（3ビット）の符号語” $xx0$ ”に変換される。

【0068】また直後の3ビット（符号語）は、変換前のデータ語において、 $m$ ビット（2ビット）単位で見て、先頭の1ビット目を同期信号を規定するために「0」とし、次の2ビット目を情報データ語とする。そして、この2ビットのデータ語を変換テーブル表3により符号語（チャネルビット）に変換する。このとき、 $m$ ビット（2ビット）のデータ語（ $0x$ ）は、 $n$ ビット（3ビット）のデータ語” $10x$ ”に変換される。

【0069】同期信号のユニークパターンを、最大ラン  $k = 7$  を破るパターンとしたとき、最も短い長さで実現できる同期信号パターンは、次に示すような12符号語（12チャネルビット）である。

” $xx0\ 100\ 000\ 000\ 10x$ ”

” $x$ ”の値は変換テーブルに依存する。上の15符号語の中には、2 (=先頭の1ビット+最後の1ビット) データ語すなわち3符号語相当を含んでいるので、実際にには12符号語が同期信号パターンのための冗長な部分である。第3チャネルビット目は、最小ランを守るために”0”が設定される。第4チャネルビット目から、同期信号パターンとして、 $k = 8$  となる9Tが設定される。すなわち”1”と”1”的間に、”0”が8個連続して並ぶ。

【0070】以上においては、同期信号パターンを最短である12符号語（12チャネルビット）として説明したが、同期信号パターンとしては、最大ラン  $k = 7$  を破る、 $k = 8$  (9T) が作成できれば良いので、12符号語以上であれば、この他にも、同期信号パターンが作ることが可能である。例えば15符号語ならば、

” $xx0\ 100\ 000\ 000\ 100\ 10x$ ”

のような同期信号を作ることができ。また、21符号語ならば、

” $100\ 000\ 000\ 10x$ ”

の”1”的個数と、変換される符号語列の要素内の”1”的個数を、2で割った時の余りが、どちらも1あるいは0で同一となるような変換規則を持っている時、上記のようにチャネルビット列内に、「反転」を表す”1”、あるいは「非反転」を表す”0”のDSV制御ビットを挿入することは、データビット列内に、「反転」するならば(1)の、「非反転」ならば(0)の、それぞれDSV制御ビットを挿入することと等価となる。

【0073】たとえば、表2において、データ変換する3ビットが(001)と続いた時に、その後ろにおいてDSV制御ビットを挿むものとすると、データは、(001-x) ( $x$ は1ビットで、「0」又は「1」)となる。ここで $x$ に「0」を与えれば、表2の変換テーブルで、

データ列 符号語列

0010 010 000

の変換が行われ、また、「1」を与えることによって

データ列 符号語列

0011 010 100

の変換が行われる。符号語列をNRZI化して、レベル符号化したとき、これらは

データ列 符号語列 レベル符号列

0010 010 000 011111

0011 010 100 011000

となり、レベル符号列の最後の3ビットが相互に反転している。このことは、DSV制御ビットxの、(1)と(0)を選択することによって、データ列内においても、DSV制御が行えることを意味する。

【0074】DSV制御による冗長度を考えると、データ列内の1ビットでDSV制御を行うということは、チャネルビット列で表現すれば、表2及び表3の変換率( $m/n = 2/3$ )より、1.5チャネルビットでDSV制御を行っていることに相当する。例えば、表1のようなRLL(1-7)テーブルにおいてDSV制御を行うためには、チャネルビット列においてDSV制御を行う必要があるが、この場合、最小ランを守るために、少なくとも2チャネルビットが必要であり、表2と表3においてデータ語でDSV制御するのに比較して、冗長度がより大きくなってしまう。換言すれば、本方式のように、データ列内でDSV制御を行うことで、効率よくDSV制御を行うことができる。

【0075】次に、図1を参照して、本発明に係る変調装置の実施の形態を図面を参考しながら説明する。この実施の形態では、データ列が、表2に従って、可変長符号(d, k; m, n; r) = (1, 7; 2, 3; 4)に変換される。

【0076】図1に示すように、変調装置1は、DSV制御ビットである「1」あるいは「0」を決定し、入力されたデータ列に、任意の間隔で挿入するDSV制御ビット決定・挿入部11、DSV制御ビットが挿入されたデータ列を変調する変調部12、並びに、変調部12の出力を記録波形列に変換するNRZI化部13を備える。また、変調装置1は、タイミング信号を生成し、各部に供給してタイミングを管理するタイミング管理部14を備える。

【0077】図2は、DSV制御ビット決定・挿入部11の処理を説明する図である。DSV制御ビットの決定及び挿入は、データ列内の任意の間隔おきに行われる。図2に示すように、入力されたデータ語のうち、まずDATA1とDATA2の間にDSV制御ビットを挿入するために、DSVビット・決定挿入部11は、DATA1までの積算DSVを計算する。DSV値は、DATA1を、チャネルビット列に変換し、さらにレベル符号化(NRZI化)した各レベルを、レベルH(1)を「+1」、レベルL(0)を「-1」として、それらの値を積算することによって得られる。同様に、

次の区間DATA2における区間DSVが計算される。次に、DATA1とDATA2の間に挿入されるDSV制御ビットx1として、DATA1、DSV制御ビットx1、およびDATA2によるDSVの絶対値が「ゼロ」に近づくような値を決定する。

【0078】DSV制御ビットx1を(1)に設定すると、DATA1の後の区間DATA2のレベル符号は反転され、また、(0)に設定すると、DATA1の後の区間DATA2のレベル符号は非反転となる。なぜならば、上記表2及び表3の各テーブル内の要素は、データ列の要素内の「1」の個数と、変換される符号語列の要素内の「1」の個数を、2で割った時の余りが、どちらも1あるいは0で一致するようになっているので、データ列内において、(1)を挿入することは、すなわち、変換される符号語列に「1」を挿入することになる(すなわち「反転」されることになる)からである。

【0079】このようにして、図2のDSV制御ビットx1が決定したら、次に所定のデータ間隔において、DATA2とDATA3の間に、DSV制御ビットx2を挿入し、同様にDSV制御を行う。なおそのときの積算DSV値は、DATA1, x1, そしてDATA2までの全てのDSV値とする。

【0080】このように、あらかじめデータ列内に、DSV制御ビットが挿入された後、変調部12で変調が行われ、チャネルビット列が発生される。

【0081】図3は、変調部12の構成例を示すブロック図である。図3において、シフトレジスタ31は、データを2ビットずつシフトさせながら、拘束長判定部32、最小ラン連続制限コード検出部33、ラン長制限補償コード検出部34、および全ての変換部35-1乃至35-4に出力するようになされている。このときシフトレジスタ31は、各部がその処理を行うのに必要なビット数を各部に供給する。

【0082】拘束長判定部32は、データの拘束長iを判定し、マルチプレクサ36に出力するようになされている。最小ラン連続制限コード検出部33は、最小ランの連続を制限する専用のコード(表2の場合、(110111))を検出したとき、その拘束長を表す検出信号(i=3)を拘束長判定部32に出力する。またラン長制限補償コード検出部34は、表2では最大ランを補償する専用のコード(表2の場合、(00001000)、または、(00000000))を検出したとき、その拘束長を表す検出信号(i=4)を拘束長判定部32に出力する。

【0083】最小ラン連続制限コード検出部33により専用のコードが検出されたとき、あるいはラン長制限補償コード検出部34により、専用のコードが検出されたとき、拘束長判定部32は、対応する拘束長をマルチプレクサ36に出力する。このとき、拘束長判定部32では、別の拘束長を判定している場合があるが、最小ラン連続制限コード検出部33またはラン長制限補償コード検出部34から専用コードによる検出出力があれば、拘

束長判定部32は、そちらを優先させて拘束長を決定する。言い換れば、より大きい拘束長が選択される。

【0084】変換部35-1乃至35-4は、内蔵されている変換テーブルを参照し、供給されたデータに対応する基礎コードが登録されているか否かを判断し、登録されている場合は、そのデータを対応する符号語に変換した後、変換後の符号語をマルチブレクサ36に出力するようになされている。また、対応するデータが変換テーブルに基礎コードとして登録されていない場合、変換部35-1乃至35-4は、入力されたデータを破棄するようになされている。

【0085】なお、この変調装置12は、表2に対応するものなので、変換部35-iとしては、拘束長i=4までのものが用意されている。すなわち、変換部としては、最大拘束長rまでの分が用意されることになる。

【0086】マルチブレクサ36は、拘束長判定部32より供給される拘束長iに対応する変換部35-iが変換した符号を選択し、その符号を、シリアルデータとして、バッファ37を介して出力するようになされている。

【0087】まだ各部の動作のタイミングは、タイミング管理部14から供給されるタイミング信号に同期して管理されている。

【0088】次に、この実施の形態の動作について説明する。

【0089】最初に、シフトレジスタ31より、各変換部35-1乃至35-4、拘束長判定部32、最小連続制限コード検出部33、およびラン長制限補償コード検出部34に、データが、2ビット単位で、それそれが判定等に必要なビット数だけ供給される。

【0090】拘束長判定部32は、例えば表2に示す変換テーブルを内蔵しており、この変換テーブルを参照して、データの拘束長iを判定し、判定結果（拘束長i）をマルチブレクサ36に出力する。

【0091】最小ラン連続制限コード検出部33は、表2に示す変換テーブルのうちの、最小ランの連続を制限する置き換えコード（表2の場合、データ（110111）と、後ろに続く符号語”010”を変換する部分）を内蔵しており、この変換テーブルを参照して、最小ランの連続を制限するコードを検出したとき、拘束長i=3の検出信号を拘束長判定部32に出力する。

【0092】またラン長制限補償コード検出部34は、表2に示す変換テーブルの中の、最大ランを守る置き換えコード（表2の場合、データ（00001000）および（00000000））を内蔵しており、この変換テーブルを参照して、最大ランを守る置き換えコードを検出したとき、拘束長i=4の検出信号を拘束長判定部32に出力する。

【0093】拘束長判定部32は、最小ラン連続制限コード検出部33から拘束長i=3の検出信号が入力され

た場合には、そのとき別の拘束長を判定していたとしても、それを選択せず、最小ラン連続制限コード検出部33の検出に対応する拘束長i（表2の例の場合、i=3）をマルチブレクサ36に出力する。同様に拘束長判定部32は、ラン長制限補償コード検出部34から拘束長i=4の検出信号が入力された場合には、そのとき別の拘束長を判定していたとしても、それを選択せず、ラン長制限補償コード検出部34の検出に対応する拘束長i（表2の例の場合、i=4）をマルチブレクサ36に出力する。

【0094】このことは、結局、各検出部33、34における拘束長の判定結果と、判定部32における拘束長の判定結果が、異なった場合には、大きい方の拘束長を最終的な拘束長として選択すればよいことを意味する。

【0095】図4は、拘束長判定部32、最小ラン連続制限コード検出部33、及びラン長制限補償コード検出部34の動作の具体例を説明している。

【0096】ラン長制限補償コード検出部34は、表2に示すテーブルの、（00001000）および（00000000）の変換部分を持ち、入力された8ビットのデータが、これと一致する場合、拘束長i=4の検出信号を拘束長判定部32に出力する。

【0097】最小ラン連続制限コード検出部33は、表2に示すテーブルの、データ（110111）と符号”010”的変換部分を持ち、入力された6ビットのデータが、（110111）であり、その後の3符号語”010”である場合、拘束長i=3の検出信号を拘束長判定部32に出力する。なお、3符号語”010”的部分を、データ変換前のデータ列で表せば、（01），（001）又は（00000）となる。従って最小ラン連続制限コード検出部33は、言い換れば、（110111）+（01/001/00000）の変換部分を持ち、入力された6ビットのデータに加えて、その後の5ビットのデータまでをさらに参照し、それらがこれらのいずれかと一致する場合（（11011101），（110111001）または（1101110000）のいずれかである場合）、拘束長i=3の検出信号を拘束長判定部32に出力する。

【0098】また拘束長判定部32は、表2に示すテーブルの変換コードを内蔵しており、入力された6ビットのデータが、（000011），（000010），（000001），あるいは（000000）のいずれかに一致する場合、拘束長i=3と判定する。また、入力された4ビットのデータが（0011），（0010），（0001）のいずれかに一致する場合、拘束長判定部32は、拘束長i=2と判定する。さらに入力された2ビットのデータが（11），（10），（01）のいずれかに一致する場合、拘束長判定部32は、拘束長i=1と判定する。

【0099】ところで、入力されたデータが例えば（0

00010) であったとき、拘束長判定部32は、拘束長*i*=3と判定する。しかしながら、始めの6ビットに加えて、さらに先の2ビットが(00) であったとき、ラン長制限補償コード検出部34によって(00001000) が検出され、拘束長*i*=4と判定される。このような時、ラン長制限補償コード検出部34からの出力信号が優先され、拘束長*i*=4と決定される。

【0.100】このようにして、表2のテーブルに従つて、最大拘束長である8ビットと、必要な部分の符号語列3ビットを参照して、全ての(1)と(0)からなるデータ列の拘束長が決定される。あるいは、符号語列を用いずに、データ列のみで拘束長を決定する場合は、最大で1.1ビットのデータを参照して、全ての(1)と(0)からなるデータ列の拘束長が決定される。

【0101】拘束長判定部32は、このようにして判定した拘束長*i*を、マルチブレクサ36に出力する。

【0102】なお、拘束長判定部32では、図4に示す順序とは逆に、拘束長の小さい方から、 $i=1, i=2, i=3, i=4$ の順番で拘束長を判定するようにしてもよい。

” 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 ”  
という符号語列（チャネルビット列）が生成される。  
【0 1 0 6】このようにして生成された符号を、例えば  
NRZI化して、レベル符号に変換すると、「1」のタイミングにおいて、その論理が反転する信号となるので、こ  
の符号語列は、  
「1 1 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 1 」  
となり、2-Tの最小反転間隔がずっと連続する。このような記録符号列は、高線密度での記録再生時には、エラ  
ーが発生し易いパターンとなる。

【0107】そこで表2に示すように、最小ランの繰り返しを制限する置き換えコード(110111)を規定

”001 000 000 010 0となり、最小ランの繰り返しが連続するのが防止される。すなわち、高線密度での記録再生時に、エラーが発生し易いパターンが取り除かれることになる。なお、この置き換え変換をした場合でも、最小ランおよび最大ランは守られている。

【0108】以上においては、変調装置1で表2を用いた場合について説明したが、表3を用いることも可能である。この場合、図3の変調部12の最小ラン連続制限コード検出部33には、表3における拘束長*i* = 4のテーブルCを与えれば良い。また、ラン長制限補償コード検出部34には、表3における拘束長*i* = 2のテーブルA、拘束長*i* = 3のテーブルB、及び拘束長*i* = 4のテーブルDを与えれば良い。

【0109】ところで、表2及び表3における、データ列と符号語列の各拘束長内では、配列の順序は異なってもよい。たとえば表2の拘束長*i*=1の部分の、

【0103】変換部35-1乃至35-4は、それぞれ各拘束長*i*に対応するテーブル（変換部35-1は、*i*=1のテーブル、変換部35-2は、*i*=2のテーブル、変換部35-3は、*i*=3のテーブル、変換部35-4は、*i*=4のテーブル）を有しており、供給されたデータに対応する変換則が、そのテーブルに登録されている場合、その変換則を利用して、供給された2×*i*ビットのデータを3×*i'*ビットの符号に変換し、その符号をマルチプレクサ36に出力する。

【0104】マルチプレクサ36は、拘束長判定部32より供給された拘束長*i*に対応する変換部35-*i*により符号を選択し、その符号をシリアルデータとして、バッファ37を介して出力する。

【0.1.0.5】ここで例えば、表2において、拘束長*i*=3の最小ランの繰り返しを制限する置き換えコード(110111)が存在しないと仮定する。このときデータとして(11011101111011101)が入力されると、その変換処理は、データ(11)(01)(11)(01)…の順に行われ、

すると、  
 $(1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1)$   
 というデータ列のうち、最初のデータ  $(1\ 1\ 0\ 1\ 1\ 1\ 0)$  が、「 $(1\ 1\ 0\ 1\ 1\ 1) + "0\ 1\ 0"$ 」に該当する  
 ので置き換え変換され、  
 $"0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0"$   
 となる。さらに次のデータ  $(1\ 1\ 0\ 1\ 1\ 1\ 0\ 1)$  が、  
 「 $(1\ 1\ 0\ 1\ 1\ 1) + "0\ 1\ 0"$ 」に該当するので置き  
 換え変換され、  
 $"0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0"$   
 となる。結局、符号語列は、

0 0 0 1 0 ...	データ	符号
i = 1 1 1		* 0 *
1 0		0 0 1
0 1		0 1 0

は、次のような配列となつても良い。

データ	符号
i = 1	* 0 *
1 1	
1 0	0 1 0
0 1	0 0 1

この場合でもデータ列の要素の「1」の個数と、符号語列の要素の「1」の個数は、それぞれ2で割った時の余りがどちらも1あるいは0で一致するようになる。

【0110】次に、本発明に係る復調装置の実施の形態を図5を参照しながら説明する。この実施の形態では、可変長符号 ( $d, k ; m, n ; r$ ) = (1, 7; 2, 3; 4) が、表2を用いてデータ列に復調される。

【0111】復調装置100は、図5に示すように、伝

送路より伝送されてきた信号、または、記録媒体より再生された信号を、復調テーブル（逆変換テーブル）に基づいて復調する復調部111、並びに、復調されたデータ列より、任意の間隔で挿入されているデータ列内のDSV制御ビットを除去し、元のデータ列を復元するDSV制御ビット除去部112を備える。バッファ113は、DSV制御ビット除去部112から入力されたシリアルデータを一旦記憶し、所定の転送レートで読み出し、出力する。タイミング管理部114は、タイミング信号を生成し、各部に供給してタイミングを管理する。

【0112】復調部111は、図6に示すように、伝送路より伝送されてきた信号、または、記録媒体より再生された信号を2値化するコンバレート部121を備える。コンバレート部121はまた、入力された信号がNRZI変調されている時（レベル符号である時）、これを逆NRZI符号化（エッジ符号化）する。拘束長判定部122は、コンバレート部121によりデジタル化された信号の入力を受け、拘束長*i*を判定する。また最小ラン連続制限コード検出部123は、コンバレート部121より入力されたデジタル化された信号から、最小ランの連続を制限するために与えられた専用のコード（表2の”001000000”）を検出し、それに対応する拘束長*i*=3の検出信号を拘束長判定部122に送る。さらにラン長制限補償コード検出部124は、コンバレート部121より入力された信号から、最大ランを補償するために与えられた専用のコード（表2の”0001000100100”，”010100100100”）を検出し、それに対応する拘束長*i*=4の検出信号を拘束長判定部122に送る。

【0113】逆変換部125-1乃至125-4は、 $n \times i$ ビットの可変長符号を、 $m \times i$ ビットのデータに逆変換するテーブル（表2の場合、*i*=1乃至4のテーブルで、変換部35-1乃至35-4の変換テーブルと実質的に同一の変換テーブル）を有している。マルチブレクサ126は、逆変換部125-1乃至125-4の出力のいずれかを、拘束長判定部122の判定結果に対応して選択し、シリアルデータとして出力する。

【0114】次に図6の復調部111の動作について説明する。伝送路より伝送されてきた信号、あるいは記録媒体より再生された信号は、コンバレート部121に入力され、コンバレートされる。コンバレート部121より出力された信号は、逆NRZI符号（”1”がエッジを示す符号）のデジタル信号となって、拘束長判定部122に入力され、表2に示す変換テーブル（逆変換テーブル）に従って、拘束長の判定処理が行われる。拘束長判定部122の判定結果（拘束長）はマルチブレクサ126に出力される。

【0115】コンバレート部121から出力されたデジタル信号は最小ラン連続制限コード検出部123にも入力される。最小ラン連続制限コード検出部123は、表

2に示す、変換テーブルのうちの、最小ランの連続を制限する置き換えコード（表2の場合、符号語”001000000”を変換する部分）を内蔵しており、この逆変換テーブルを参照して、最小ランの連続を制限するコード”001000000 not100”を検出したとき、拘束長*i*=3の検出信号を拘束長判定部122に出力する。

【0116】さらに、コンバレート部121から出力されたデジタル信号はラン長制限補償コード検出部124にも入力される。ラン長制限補償コード検出部124は、表2に示す、変換テーブルの中の、最大ランを守る置き換えコード（表2の場合、符号語列”0001000100100100”、および”010100100100”）を内蔵しており、この逆変換テーブルを参照して、最大ランを守る置き換えコードを検出したとき、拘束長*i*=4の検出信号を拘束長判定部122に出力する。

【0117】入力された変調符号の判定処理についてまとめる、図7に示すようになる。すなわち、ラン長制限補償コード検出部124は、表2に示すテーブルの、”0001000100100100”、あるいは”010100100100”的逆変換部分を持ち、入力された12ビットの符号語列が、これと一致する場合、拘束長*i*=4の検出信号を拘束長判定部122に出力する。

【0118】最小ラン連続制限コード検出部123は、表2に示すテーブルの、”001000000”的逆変換部分を持ち、入力された12ビットの符号語列が、”001000000 not100”と一致する場合、拘束長*i*=3の検出信号を拘束長判定部122に出力する。なお、拘束長の判定には特に必要ないが、入力された符号語列を12ビット分見ると、このときの符号語は、”001000000010”となっている。

【0119】また拘束長判定部122は、表2に示す逆変換テーブルを内蔵しており、入力された9ビット又は12ビットの符号語列が、”0001000100”，”000100000 not100”，”010100100100”，あるいは”010100000 not100”的いずれかに一致する場合、拘束長*i*=3と判定する。これに当てはまらない場合、入力された6ビット又は9ビットの符号語列が、”010100”，”0100000 not100”，あるいは”000100”的いずれかに一致するとき、拘束長判定部122は、拘束長*i*=2と判定する。さらにこれに当てはまらない場合、入力された3ビットの符号語列が、”000”，”101”，”001”，あるいは”010”的いずれかに一致するときに、拘束長判定部122は、拘束長*i*=1と判定する。

【0120】なお、拘束長判定部122、最小ラン連続

制限コード検出部123、及びラン長制限補償コード検出部124の拘束長判定の処理は、拘束長の小さい方から、 $i=1, i=2, i=3, i=4$ の順番で行うようにしてもよい。

【0121】拘束長を、その小さい方から、 $i=1, i=2, i=3, i=4$ の順番で判定していった場合、入力された符号語列が例えば、"000 100 100 100"であったとき、拘束長判定部122において、拘束長の小さいほうから順に、一致または不一致を判定していくと、拘束長  $i=1$  あるいは、拘束長  $i=2$ 、拘束長  $i=3$ 、そして拘束長  $i=4$  と、全ての拘束長にあてはまることがある。このような場合は、決定規則どして、それぞれ判定された拘束長から最大のものを選択し、決定すればよい。

【0122】逆変換部125-1乃至125-4のうち、例えば逆変換部125-1には、アドレス"10

<表4> 逆変換テーブル (1; 7; 2; 3; 4)

符号語列	復調データ列
i=1 101	11
000	11
001	10
010	01
i=2 010 100	0011
010 000(not 100)	0010
000 100	0001
i=3 000 100 100	000011
000 100 000(not 100)	000010
010 100 100	000001
010 100 000(not 100)	000000
i=3 : Prohibit Repeated Minimum Transition Runlength	
001 000 000(not 100)	110111
i=4 : limits k to 7	
000 100 100 100	00001000
010 100 100 100	00000000

【0126】次に、図8のフローチャートを参照して、DSV制御ビット除去部112の動作について説明する。DSV制御ビット除去部112は、内部にカウンタを有しており、ステップS1において、復調部111よりデータ列のビットが入力されると、その数をカウントする。ステップS2において、カウント値がDSV制御ビットを挿入する所定のデータ間隔に達したか否かが判定され、任意のデータ間隔ではないと判定された場合、ステップS3において、復調部111より入力されたデータがそのままバッファ113に出力される。これに対して、ステップS2において、所定のデータ間隔であると判定された場合、そのビットはDSV制御ビットであるから、ステップS3の処理はスキップされる。すなわち、この場合には、そのビットはバッファ113に出力されず、廃棄される。

"1" および "000" にデータ (11) が、アドレス "001" にデータ (10) が、そしてアドレス "010" にデータ (01) が、それぞれ書き込まれている。以下、逆変換部125-2乃至125-4の各逆変換テーブルも、同様に、それぞれ対応するデータが書き込まれており、供給された  $3 \times i$  ビットの符号語列を、 $2 \times i$  ビットのデータ列に変換し、そのデータ語をマルチブレクサ126に出力する。

【0123】マルチブレクサ126は、逆変換部125-1乃至125-4より供給されたデータのいずれかを、拘束長判定部122の拘束長判定結果に対応して選択し、シリアルデータとして出力する。

【0124】表2の逆変換テーブルを示すと、次の表4のようになる。

【0125】

【0127】次に、ステップS4に進み、次のデータを入力する処理が実行される。そして、ステップS5において、全てのデータに対する処理が終了したか否かが判定され、まだ処理していないデータが存在する場合には、ステップS1に戻り、それ以降の処理が繰り返し実行される。ステップS5において、全てのデータを処理したと判定された場合、処理は終了される。

【0128】以上のようにして、DSV制御ビット除去部112より出力されるデータからは、DSV制御ビットが除去されることになる。このデータは、バッファ113を介して出力される。

【0129】以上においては、復調装置100に表2の変換テーブル（表4の逆変換テーブル）を用いた場合について説明したが、表3の変換テーブル（表5に示す表3に対応する逆変換テーブル）を用いた場合にも、同様

の処理を実行することができる。この場合、図6の最小ラン連続制限コード検出部123には、表3における拘束長*i*=4のテーブルCを与えるべき。また、ラン長制限補償コード検出部124には、表3における拘束長*i*=2のテーブルA、拘束長*i*=3のテーブルB、及び拘束長*i*=4のテーブルDを与えるべき。

## 【0130】

&lt;表5&gt;

逆変換テーブル(1, 7; 2, 3; 4)

符号語列 複調データ列

r=1 Main table:

101	00
100	01
001	10
000	11

r=2 substitution table A.(limits d to 1)

100 010	0000
101 010	0001
000 010	1000
001 010	1001

r=3 substitution table B.(limits k to 8)

000 010 010	111111
001 010 010	111110
101 010 010	011110
100 010 010	011111

r=4 substitution table C.(limits RMTR to 6)

100 010 010 010	00010001
100 000 010 010	10010001
000 010 010 010	10010001

r=4 substitution table D.(limits k to 7)

000 001 010 010	11100000
100 001 010 010	11100010
001 010 010 010	11100001
101 010 010 010	11100000

【0131】ところで、データに同期信号(Sync)

”#01 001 000 000 001 000 000 001”

と決定する。”#”は、同期信号の挿入により区切られた、直前のデータ列(DSV制御ビットは含んで良い)に依存しており、区切られたデータ列を変換テーブルに従って変換した際に、終端テーブルを用いて終端させた場合には

”#”=”1”

とされ、また終端テーブルを用いずに、表2のテーブルにより終端した場合には

”#”=”0”

とされる。変調部12は、終端テーブルを用いた場合には、”#”=”1”を、用いない場合には、”#”=”0”を、同期信号決定部211に出力する。同期信号決定部211は、変調部12から、この”#”の値の入力を受けると、これを同期信号の先頭ビットに挿入する。

を挿入する必要がある場合がある。次に、この場合の変調装置1と復調装置100について、図9と図10を参照して説明する。これらの実施の形態でも、データ列が、表2に従って、可変長符号(d, k; m, n; r)=(1, 7; 2, 3; 4)に変調され、また復調されるものとする。

【0132】所定の間隔で同期信号を挿入する変調装置1においては、図9に示すように、DSV制御ビット決定・挿入部11の出力が、同期信号決定部211に供給される。同期信号決定部211にはまた、変調部12の出力も供給されている。同期信号決定部211は、入力された信号から同期信号を決定すると、その出力を同期信号挿入部212に出力している。同期信号挿入部212は、変調部12より入力される変調信号に、同期信号決定部211より入力される同期信号を挿入し、NRZI化部13に出力している。その他の構成は、図1における場合と同様である。

【0133】同期信号決定部211は、同期信号パターンを24符号語とするとき、表2に従って、同期信号を、

そして、その同期信号を同期信号挿入部212に出力する。

【0134】同期信号挿入部212は、同期信号決定部211から入力される同期信号を、変調部12の出力に挿入し、NRZI化部13に出力する。その他の動作は、図1における場合と同様である。

【0135】同期信号が挿入された後の最初のデータは、その先頭から(同期信号の直前のデータを考慮することなく)変換処理される。変調部12、および同期信号決定部211は、同期信号が挿入される所定の間隔をカウントするためのカウンタを備え、そのカウント値に対応して、同期信号の位置を決定する。

【0136】なお、図9の例では表2の変換テーブルを用いるようにしたが、表3の変換テーブルを用いること

も可能である。この場合、例えば図9における同期信号決定部211は、表3の同期信号パターンとして、12符号語の同期信号を採用するとき、

”**x x 0 1 0 0 0 0 0 0 0 1 0 x**”

を同期信号とする。”x”は、同期信号挿入により区切られた、直前及び直後のデータ列（DSV制御ビットは含んで良い）に依存しており、先頭の3符号語と最後の3符号語は、表3により決定される。すなわち同期信号の挿入により区切られた最後のデータ列を（p）、また、その直後の先頭のデータ列を（q）とすると、（p1）として、表3を用いて変換し、その後に”**1 0 0 0 0 0 0 0 0**”を挿み、最後に（0q）として、表3を用いて変換する。こうすることで、同期信号として必要な最大ランkを破る、k=8（9T）を必ず発生することができる。

**【0137】**図10は、図9の変調装置1により変調された符号を復調する復調装置100の構成例を表している。この例においては、所定の伝送経路を介して入力された符号が、復調部111と同期信号識別部221に入力されている。同期信号識別部221は、入力された符号と復調部111から入力された信号を用いて、同期信号を識別し、識別信号を同期信号除去部222に出力している。同期信号除去部222は、復調部111から入力された復調信号から、同期信号識別部221の出力に対応して同期信号を除去し、同期信号を除去した信号をDSV制御ビット除去部112に出力している。その他の構成は、図5における場合と同様である。

**【0138】**同期信号識別部221は、内蔵するカウンタで符号語をカウントし、そのカウント値から所定の間隔で挿入されている同期信号の位置を決定する。同期信号パターンの位置が判明したとき、同期信号識別部221は、次に変調時に定めた”#”の部分を読み取る。即ち同期信号ビット部分の先頭ビットを読み取り、それを復調部111に出力する。復調部111は、先頭ビットが”1”であれば、その直前の符号の復調には、表2の終端テーブルを用いる。また先頭ビットが”0”であれば、復調部111は、その直前の符号の復調には、表2の変換コードのテーブルを用いる。これ以外の同期信号ビットは、情報を持たないビットであるから不要となる。

**【0139】**同期信号識別部221は、同期信号を構成するビットを識別する識別信号を同期信号除去部222に出力する。同期信号除去部222は、復調部111より入力されたデータから、この識別信号によって指定された同期信号ビットだけを除去し、DSV制御ビット除去部112に出力する。

**【0140】**なお、図10では表2の変換テーブルを用いるようにしたが、表3の変換テーブルを用いることもできる。この場合、例えば図10における同期信号識別部221は、所定の間隔で挿入されている同期信号の位

置をカウンタの値から決定する。同期信号パターンの位置が判明したとき、同期信号識別部221は、同期信号パターンの先頭の3符号語、及び、最後の3符号語を指定する信号を復調部111に出力する。これらの符号語には、データ列が含まれているので、これを含めるように復調部111で復調が行われる。

**【0141】**同期信号識別部221は、同期信号のデータ部分を除くユニークパターンの部分のビットを指定する信号を同期信号除去部222に出力する。同期信号除去部222は、この信号に対応して同期信号ビット（ユニークパターンのビット）だけを除去する。

**【0142】**図11に同期信号とDSV制御ビットを挿入した記録符号列の例を示す。この例では、同期信号として24符号語が用いられ、DSV制御は56データビット置きに行なわれ、5回のDSV制御ごとに同期信号が挿入されている。このとき、同期信号ごとの符号語数（チャネルビット数）は、  

$$24 + (1+56+1+56+1+56+1+56+1+56+1) \times 1.5 = 453$$
 符号語（チャネルビット）  
 となる。このときのデータ語の冗長度は、次のように、約7.3%となる。

$$(56 \times 5) \times 1.5 / 453 = 420 / 453 = 0.927$$

$$1 - 0.927 = 0.0728$$

**【0143】**発明者等は、以上の変換テーブルを用いた変調結果をシミュレーションしてみた。Tminの連続を制限し、かつデータ列内においてDSV制御ビットを挿入したデータ列を変調した結果について以下に示す。シミュレーションには、表2及び表3が用いられた。さらに比較のために、従来のRLL(1-7)変調である表1についてもシミュレーションが行われた。

**【0144】**任意に作成したランダムデータ13107200ビットを、56データビットおきにDSV制御ビットを1ビットを挿入することでDSV制御した後、表2または表3の変換コードテーブルを用いて、符号語列（チャネルビット列）に変換した場合の結果は以下の通りである。また同様に、任意に作成したランダムデータ13107200ビットを、表1の変換コードテーブルを用いて、符号語列（チャネルビット列）に変換し、さらに112符号語（チャネルビット）おきに、DSV制御ビットとして2チャネルビットを挿入することでDSV制御を行った時の結果は以下の通りである。

**【0145】**ここで、表2、表3では56データビットおき、また、表1では112符号語おきとしたのは、DSV制御ビットの冗長度を同一にするためである。このように、DSV制御の必要ビット数に差がある場合には、冗長度をそろえて考えた時、効率良くDSV制御が行える表2や表3の方が、表1に較べて低域特性が良好となる。

**【0146】**また、各結果の数値は以下のようにして計

算した。

**Ren\_cnt[1 to 10]:** 最小ランの繰り返し1回乃至10回の各発生数。

**T\_size[2 to 10]:** 2T乃至10Tの各ランの発生数。

**Sum : Number of bits.** ビット総数。

**Total: Number of runlengths.** 各ラン(2T, 3T, ...)の発生総数

**Average Run : (Sum/Total)**

ラン分布の数値:  $(T\_size[i] * (i)) / (Sum)$ ,  $i = 2, 3, 4, \dots, 10$

表6の2T乃至10Tの欄に示す数値が、このラン分布の数値を表す。

最小ランの連続する分布の数値:  $(Ren\_cnt[i] * (i))$

$\vee T\_size[2T],$   
 $i=1, 2, 3, 4, \dots, 10$

表6のRMTR(1)乃至RMTR(9)の欄に示す値が、この最小ラ

<表6>

\*\*\* PP17 comparison \*\*\*

	<表2> 17PP-32	<表3> 17PP-52	<表1> +2bit-DC (Without-DCC) (DSV制御) (DSV制御なし)	
Average Run	3.3665	3.4048	3.3016	3.2868
Sum	20011947	20011947	20011788	19660782
Total	5944349	5877654	6061150	5981807
2T	0.2256	0.2246	0.2417	0.2419
3T	0.2217	0.2069	0.2234	0.2281
4T	0.1948	0.1935	0.1902	0.1915
5T	0.1499	0.1491	0.1502	0.1511
6T	0.1109	0.1094	0.1135	0.1141
7T	0.0579	0.0814	0.0561	0.0544
8T	0.0392	0.0351	0.0218	0.0188
9T	-----	-----	0.0023	-----
10T	-----	-----	0.0009	-----
RMTR(1)	0.3837	0.3890	0.3628	0.3641
RMTR(2)	0.3107	0.3137	0.2884	0.2883
RMTR(3)	0.1738	0.1906	0.1717	0.1716
RMTR(4)	0.0938	0.0806	0.0909	0.0907
RMTR(5)	0.0299	0.0228	0.0456	0.0452
RMTR(6)	0.0081	0.0033	0.0219	0.0217
RMTR(7)	-----	-----	0.0100	0.0099
RMTR(8)	-----	-----	0.0047	0.0046
RMTR(9)	-----	-----	0.0022	0.0022
max-RMTR	6	6	18	18
peak DSV	# -36 to 36	# -35 to 40	* -46 to 43	* -1783 to 3433
	(*#*: 56data-bit+1dc-bit, 1.75%)		(*#: 112cbit+2dc-cbit, 1.75%)	

【0148】上の結果より、表2及び表3を用いると、RLL(1, 7)方式となっていること、最小ランと最大ランが守られ、かつ最小ランの連続が最大で6回までに

ンの連続する分布の数値を表す。

**max-RMTR:** 最小ランの繰り返す、最大回数。

**peak DSV:** チャネルビット列のDSV制御を行う過程において、DSV値を計算したときのDSV値のプラス側のピーク及びマイナス側のピークを言う。

DSV制御ビットとして56データ列おきにDSV制御ビットを挿入した場合の冗長率は、56データ列に対してDSV制御ビット1ビットであるから、次のようになる。

$$1 / (1 + 56) = 1.75\%$$

また、DSV制御ビットとして112符号語列おきに2ビットのDSV制御ビットを挿入した場合の冗長率は、112符号語列に対してDSV制御ビット2ビットであるから、次になる。

$$2 / (2 + 112) = 1.75\%$$

すなわち同じ冗長率である。

【0147】

制限されていることが確認された。また、peak DSVの結果より、データ列内でDSV制御を行うことができる(peak DSVの値が所定の範囲内に納められている)こと、こ

の場合、DSV制御ビットの効率がよいで、従来の符号語列（チャネルビット列）にDSV制御ビットを挟む方法よりも良好な低域成分を得ることができる（peak DSVの振れ幅が、表1の場合、89（=46+43）であるのに対して、表2の場合、72（=36+36）、表3の場合、75（=35+40）と、表1の場合の値より小さくなっている）ことが確認された。

【0149】以上より、従来のRLL（1-7）方式（表1の方式）と比較すると、表2または表3を用いる方式（17PP方式）は、最小ランの繰り返しが多くても6回までに制限することができる、高線密度におけるエラー特性の改善を見込むことが出来る。

【0150】また、DSV制御の効率が良いので、同じ冗長度1.75%でDSV制御を行うと、17PP方式の方がピークDSV値の差が小さく出来、したがって、より低域成分を抑制することが可能となるので、安定したデータ記録再生を行うことが出来る。

<表7>  
Shift error response

	<表2>	<表3>	<表1>
worst case	17PP-32	17PP-52	+2bit-DC
(dc bit)	3 Bytes	3 Bytes	2 Bytes
Byte error(0)	in.	in.	without
Byte error(1)	0.028	0.096	0.080
Byte error(2)	0.777	0.635	0.826
Byte error(3)	0.195	0.268	0.094
Average -			-----
Byte error rate	1.167Byte	1.174Byte	1.014Byte

【0153】以上のように、この実施の形態は、最小ランd=1、最大ランk=7、変換率m/n=2/3の変換テーブルにおいて、最小ラン長の繰り返し回数を制限する置き換えコードを設けるようにしたので、

(1) 高線密度での記録再生、及び、タンジェンシャル・チルトに対する許容度が向上する。  
(2) 信号レベルが小さい部分が減少し、AGCやPLL等の波形処理の精度が向上し、総合特性を高めることができる。

(3) 従来と比較して、ビタビ復号等の際のバスメモリ長を短く設計することができ、回路規模を小さくすることができる。

【0154】また、変換テーブルの要素内の「1」の個数と、変換される符号語列の要素内の「1」の個数を、2で割った時の余りが、どちらも1あるいは0で一致するようにしたので、

(4) DSVの制御のための冗長ビットを少なくすることができる。  
(5) 最小ランd=1かつ(m,n)=(2,3)においては、1.5符号語でDSV制御を行うことができる。  
(6) 冗長度が少ない上に、最小ランと最大ランを守る

【0151】さらに、シミュレーションにおいて、上述した場合と同一のランダムデータを使って発生させたチャネルビット列における、ビットシフト時の復調エラー伝搬特性を調べたところ、17PPの最悪エラー伝搬は、3バイトであるが、実際の発生頻度はほとんどないことが確認され、従来のRLL（1-7）に較べての悪化はそれほどではない（平均バイトエラーレートは、表1の場合、1.014バイトであるのに対して、表2の場合、1.167バイトであり、表3の場合、1.174バイトである）ことが確認された。なお、このエラーレートの結果の数値は、本発明によるテーブルではDSV制御ビットを含み、また従来RLL（1-7）では含んでいない。すなわち必ずしも同じ条件での測定とは言えず、これらが数値に影響を及ぼすことが考えられ、比較にあたってはその点を考慮する必要がある。

【0152】

ことができる。

【0155】さらに本テーブルは特に、ラン長制限を守る置き換えコードを設けるようにしたので、

(7) テーブルがコンパクトになる。

(8) ビットシフト時の復調エラー伝搬を、表1の場合と殆ど同じ状態にできる。

【0156】なお、上記したような処理を行うコンピュータプログラムをユーザーに提供する提供媒体としては、磁気ディスク、CD-ROM、固体メモリなどの記録媒体の他、ネットワーク、衛星などの通信媒体を利用することができる。

【0157】

【発明の効果】以上の如く、請求項1に記載の変調装置、請求項2-5に記載の変調方法、請求項2-6に記載の提供媒体、請求項2-7に記載の復調装置、請求項3-0に記載の復調方法、および請求項3-1に記載の提供媒体によれば、データ列の要素内の「1」の個数と、変換される符号語列の要素内の「1」の個数を、2で割った時の余りが、どちらも1あるいは0で一致するような変換規則、最小ランdの連続を有限回以下に制限する第1の置き換えコード、およびラン長制限を守るための第2の置

き換えコードを有する変換テーブルで変換処理を行うようにしたので、少ない冗長度でDSV制御を行うことができるとともに、高線密度でエラーの少ない符号語列を記録再生することが可能となり、さらに、ビットシフト時の復調エラー伝搬の増加を抑制することができる。

【図面の簡単な説明】

【図1】本発明の変調装置の構成例を示すブロック図である。

【図2】図1のDSV制御ビット決定・挿入部11の動作を説明する図である。

【図3】図1の変調部12の構成例を示すブロック図である。

【図4】図3の変調部12の動作を説明する図である。

【図5】本発明の復調装置の構成例を示すブロック図である。

【図6】図5の復調部111の構成例を示すブロック図である。

【図7】図6の復調部111の動作を説明する図である。

【図8】図5のDSV制御ビット除去部112の処理を説明するフローチャートである。

【図9】本発明の変調装置の他の構成例を示すブロック図である。

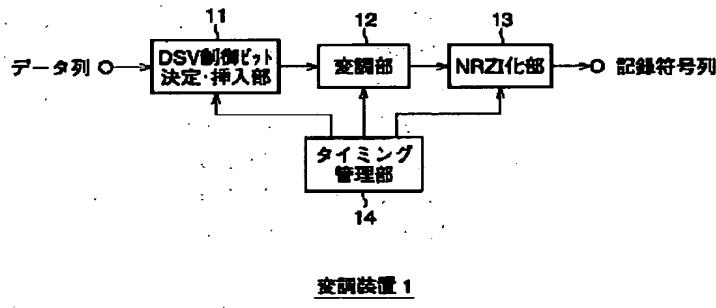
【図10】本発明の復調装置の他の構成例を示すブロック図である。

【図11】同期信号とDSV制御ビットを挿入した記録符号列の例を示す図である。

【符号の説明】

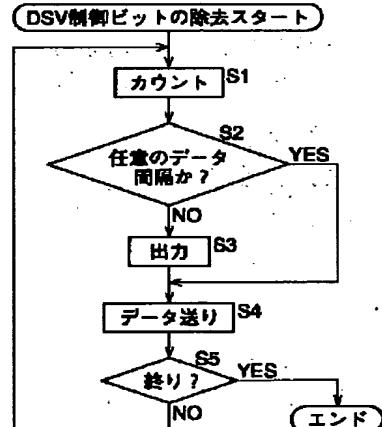
11 DSV制御ビット決定・挿入部, 12 変調部,  
13 NRZI化部, 31 シフトレジスタ, 32 拘束長判定部, 33 最小ラン連続制限コード検出部,  
34 ラン長制限補償コード検出部, 35-1 乃至  
35-4 変換部, 36 マルチプレクサ, 37  
バッファ, 111 復調部, 112 DSV制御ビット除去部, 121 コンパレート部, 122 拘束長判定部, 123 最小ラン連続制限コード検出部,  
124 ラン長制限補償コード検出部, 125-1 乃至  
125-4 逆変換部, 126 マルチプレクサ

【図1】

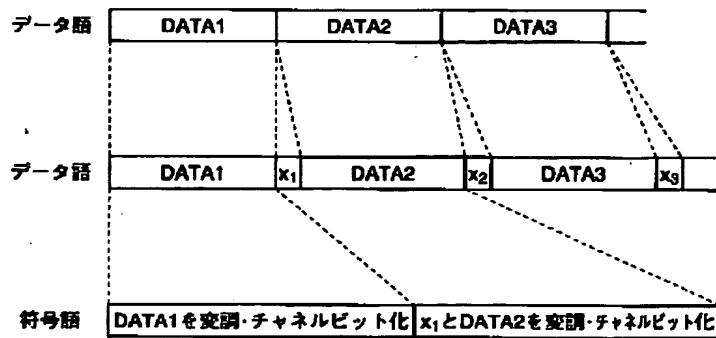


変調装置1

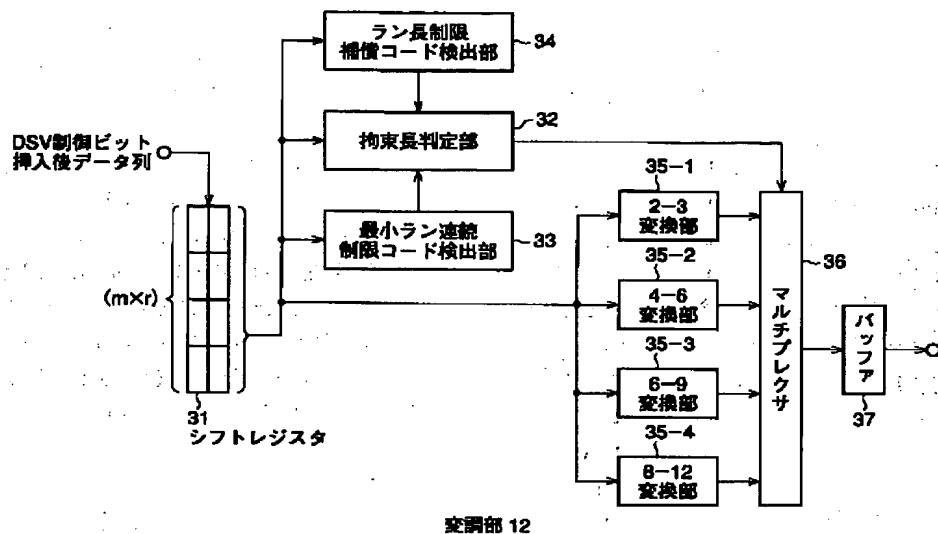
【図8】



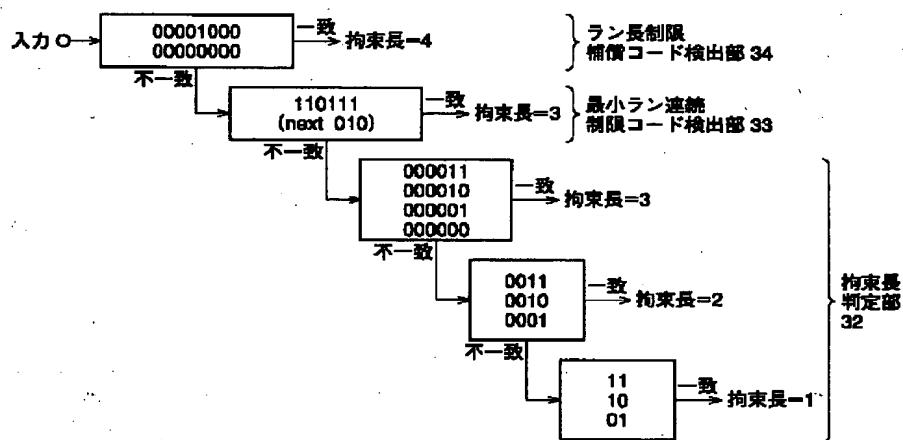
【図2】



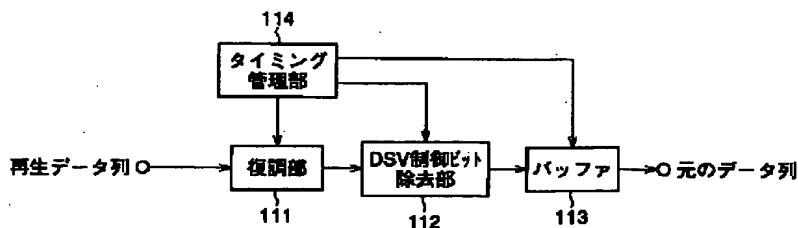
【図3】



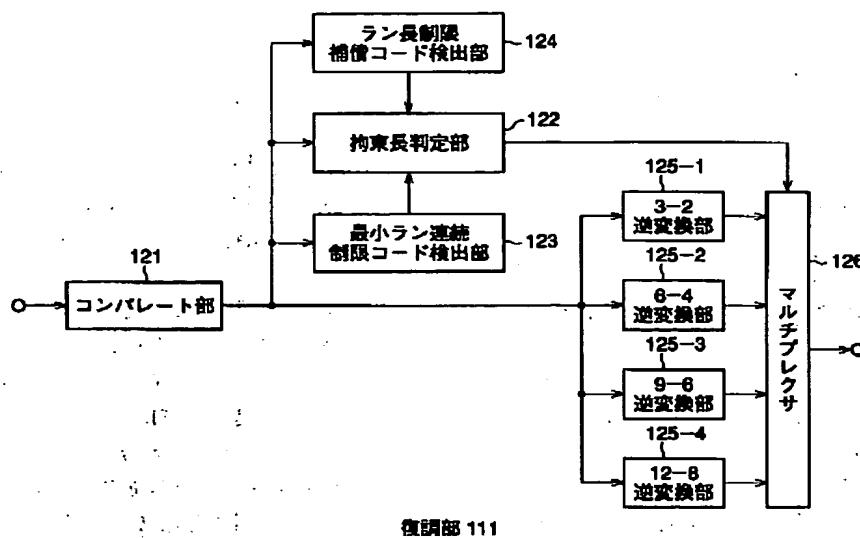
【図4】



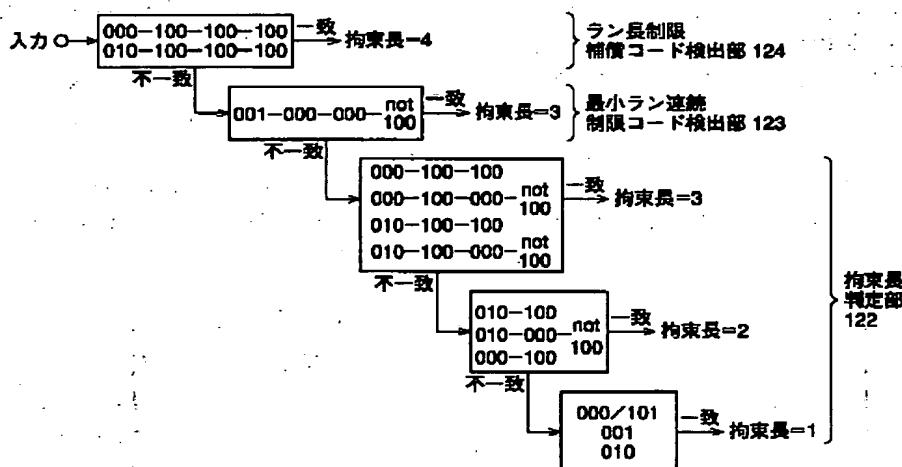
【図5】

復調装置 100

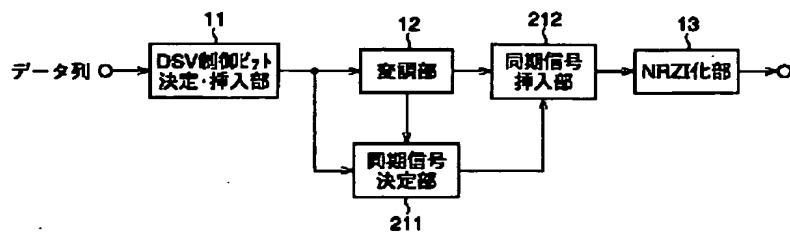
【図6】



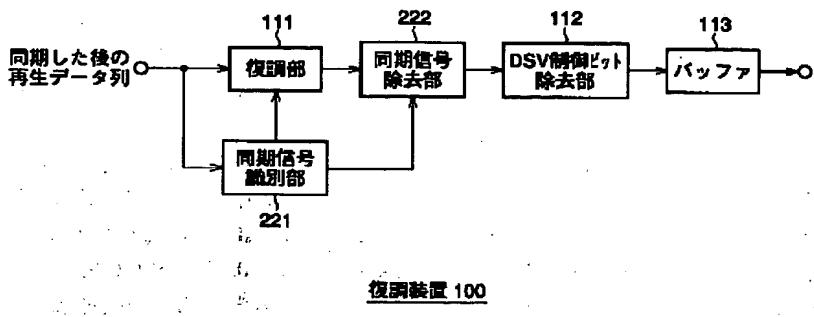
【図7】



【図9】

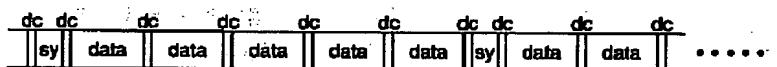


【図10】



### 【図1-1】

dc : DSV制御ビット  
sy : 同期信号



## フロントページの続き

(71)出願人 598070935

Groenewoudseweg 1  
5621 BA Eindhoven The  
Netherlands

(72)発明者 中川 俊之  
東京都品川区北品川6丁目7番35号 ソニ

(72)発明者 新福 吉秀  
東京都品川区北品川6丁目7番35号 ソニ  
株式会社

(72)発明者 楠原 立也  
東京都品川区北品川6丁目7番35号 ソニ  
株式会社

(72)発明者 中村 耕介  
東京都品川区北品川6丁目7番35号 ソニ  
一株式会社内

(72)発明者 ケース・エイ・スハウハーメル・イミンケ  
オランダ アイントホーフェン市・エイ・  
エイ5656 ホルスト教授通り4 (ダブリュ  
ー・ワイ61) 株式会社オランダ・フィリ  
ップス企業グループフィリップス中央研究  
所内

(72)発明者 ジー・ジェイ・バン・デン・エンデン  
オランダ アイントホーフェン市 エイ・  
エイ5656 ホルスト教授通り4(ダブリュ  
ー・ワイ61) 株式会社オランダ・フィリ  
ップス企業グループフィリップス中央研究  
所内

(72)発明者 ジェイ・エイ・エッチ・エム・カールマン  
オランダ アイントホーフエン市 エイ・  
エイ5656 ホルスト教授通り4 (ダブリュ  
ー・ワイ61) 株式会社オランダ・フィリ  
ップス企業グループフィリップス中央研究  
所内

# PATENT ABSTRACTS OF JAPAN

(11) Publication number : 11-346154  
 (43) Date of publication of application : 14.12.1999

(51) Int.CI. H03M 7/14

(21) Application number : 10-150280

(71) Applicant : SONY CORP  
 PHILIPS ELECTRONICS NV

(22) Date of filing : 29.05.1998

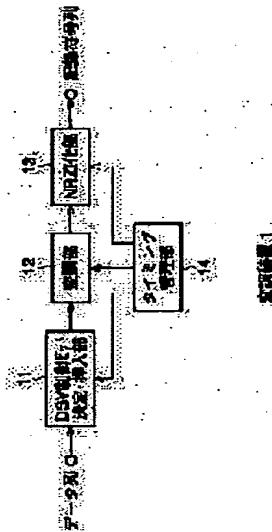
(72) Inventor : NAKAGAWA TOSHIYUKI  
 NIIFUKU YOSHIHIDE  
 NARAHARA TATSUYA  
 NAKAMURA KOSUKE  
 KEESU A SUHAUHAAMERU IMINKU  
 G J VAN DEN ENDEN  
 J A H M KARLMANN

## (54) DEVICE AND METHOD FOR MODULATION, DEVICE AND METHOD FOR DEMODULATION AND PROVIDING MEDIUM

### (57) Abstract:

**PROBLEM TO BE SOLVED:** To record and to reproduce at high linear density.

**SOLUTION:** A DSV control bit decision/insertion part 11 inserts a DSV control bit for DSV control in an inputted data sequence and outputs it to a modulation part 12. The modulation part 12 converts the data of two bit basic data length into a variable length code of three bit basic code length and outputs them to a non return to zero inverted(NRZI) part 13 in accordance with a conversion table. A conversion table which the modulation part 12 possesses has a replacement code for restricting continuation of a minimum run below the specified number of times and a replacement code for protecting a run length restriction, and moreover, a conversion rule which has a remainder after the number of '1' in an element of the data sequence is divided, by 2, and the remainder after the number of '1' in an element of a code word sequence is divided by 2 agree with each other as 1 or 0.



### LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

**THIS PAGE BLANK (USPTO)**